WEB

ES6: event-driven programming

+0.1

Stack 31
X
25

querySelector

First we need to access the button element in our JavaScript code.

variable name. We store the button in theis variable to reuse it later

selects the first HTML element which the query applies to

```
const button = document.querySelector(“button”);
```

variable defination, we could use let or var instead. But const id preferred

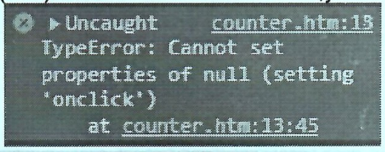
The context in which the querySelector will search for the given query. Here this is the HTML file

same syntax as CSS selectors. you can refer to elements, ids or classes here.

```
<html>
<head>
  <title>Hello</title>
  <script>
  function Hello(){
document.querySelector('h2').innerHTML=
                    "Hello JavaScript!";}
  </script>
</head>
<body>
<h2></h2>
<button onclick="Hello()">Click
Me!</button>
</body>
</html>
```

```
document.addEventListener
('DOMContentLoaded',function(){
document.querySelector('button').onclick
                        = Hello; }
);
```

```
<html>
<head>
  <title>Hello</title>
</head>
  <script>
  function Hello(){
document.querySelector('h2').innerHTML="Hello JS!";}
   </script>
  <body>
  <h2></h2>

  <script>
   document.querySelector('button').onclick =
      Hello;
</script>
```

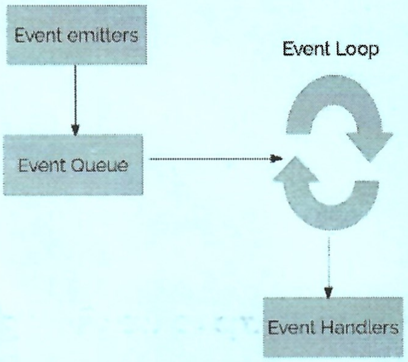


```
Uncaught              counter.htm:13
TypeError: Cannot set
properties of null (setting
'onclick')
    at counter.htm:13:45
```

<button ~~onclick="Hello()"~~>Click Me!</button>

```
<script>
  document.querySelector('button').onclick = Hello;
</script>
</body>
</html>
```

document.querySelector('button').addEventListener('click',Hello);



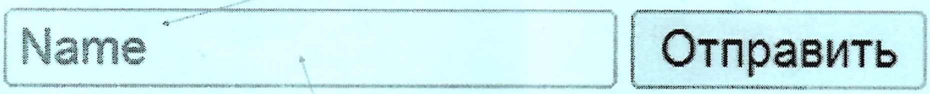
```
<button onclick="Hello()">Click
Here</button>
<script>
let counter = 0;
function Hello(){
      counter++;
      const heading=
        document.querySelector('h1');
      heading.innerHTML = counter;
   }
   </script>
```

HTML forms

The placeholder - the thing the user sees filled into that input field originally

automatically focus this input field

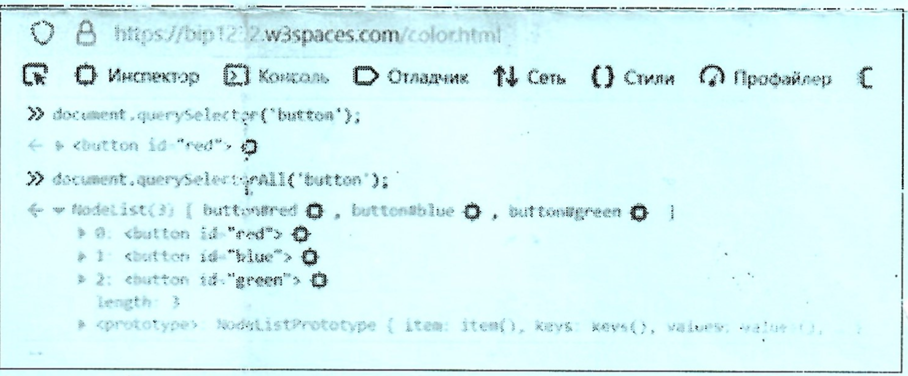
```
<form>
    <input autofocus id="name" placeholder="Name" type="text">
    <input type="submit">
</form>
```



input field where the user can type in some text

```
<script>
    document.addEventListener('DOMContentLoaded', function()
{document.querySelector('form').onsubmit = function() {
   let name = document.querySelector('#name').value;
                    alert(`Hello, ${name}!`);
                };
            });
 </script>
```

```
<button
id="red">Red
</button>
<button
id="blue">Blue
</button>
<button
id="green">Green
</button>
```



```
>> document.querySelector('button');
← ▸ <button id="red">
>> document.querySelectorAll('button');
← ▾ NodeList(3) [ button#red , button#blue , button#green ]
      ▸ 0: <button id="red">
      ▸ 1: <button id="blue">
      ▸ 2: <button id="green">
        length: 3
      ▸ <prototype>: NodeListPrototype { item: item(), keys: keys(), values: value(), … }
```

```
document.addEventListener('DOMContentLoaded',function(){
document.querySelector('button').onclick = Hello; }
);
document.querySelector('#red').onclick = function() {
   document.querySelector('#hello').style.color = 'red'; };
```

Web

+0.1 2025.X.23

ES6:

asp.net.by/es6/1.htm, asp.net.by/es6/02.htm

```
{
 let
 const
}
```

asp.net.by/es6/03.htm **Arrow functions:**

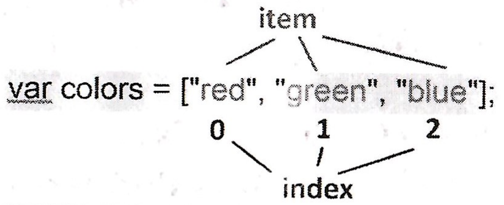
```
x = (x, y) => y + ":" + x;
                                   x("sbmt.by", "SBMT");
```

asp.net.by/es6/04.htm function myExam(y = 4) – **параметр по умолчанию**

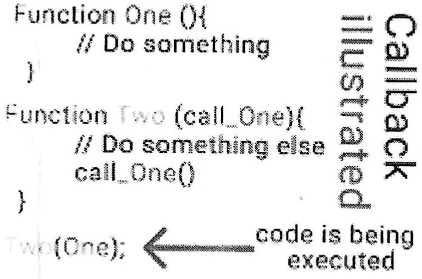
asp.net.by/es6/05.htm Array.**find**(myFunction)
asp.net.by/es6/06.htm Array.**findIndex**()
asp.net.by/es6/07.htm Классы

```
class ACat {

  constructor(n) {

    this.name = n;      //property

  }

}

mycat = new ACat("Барсик");
```

```
class ACat{
......................................
 Name()
   return 'Меня зовут ' + this.name;
 }
}
mycat = new ACat("Барсик");
document.write(mycat.Name());
```

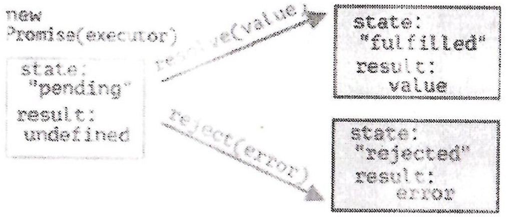
asp.net.by/es6/08.htm asp.net.by/es6/09.htm Array.**forEach**() var colors = ["red", "green", "blue"];	Меня зовут Барсик  var colors = ["red", "green", "blue"];

```
colors.forEach(myFunction);
function myFunction(item, index) {
   item … index
}
```



```
Function One (){
      // Do something
 }
Function Two (call_One){
      // Do something else
      call_One()
 }
Two(One);
```

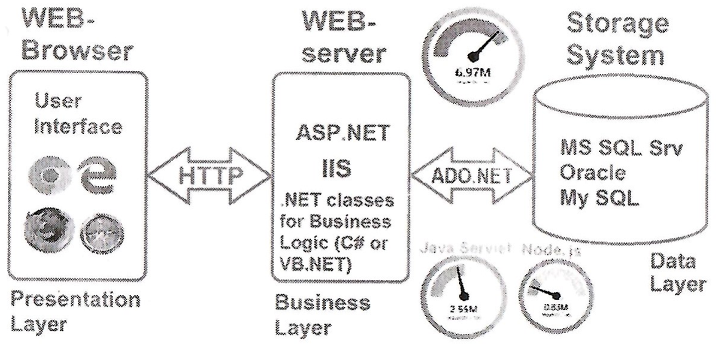
Promise



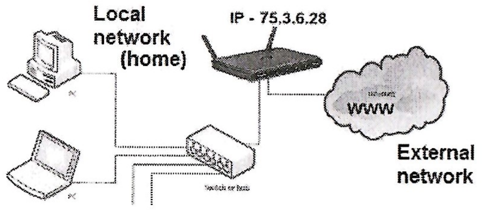
+0.1 to Fin 15.9.25.

+01 2025.9.24

WEB



Front End
HTML/
CSS/ Java
Script и
DOM
BackEnd
ASP.NET
IIS
ADO.NET



DHCP (engl. Dynamic Host Confi-guration Protocol)

tracert – tracerout (in Apple) www.geotraceroute.com

172 . 16 . 254 . 1

10101100.00010000.11111110.00000001

8 bits

32 bits (4 bytes)

4 billion addresses – **4.294.967.296**

DNS – clearer address.

217.21.43.40 – sbmt.by, sbmt.bsu.by sb.bsu.by

ping sb.bsu.by tracert sb.bsu.by https://geotraceroute.com/

+375 (29) 254 07 92 - ANDREY O. YAROSHEVICH

Request

Method URL Protocol version

Header

GET/ index.htm HTTP/1.1
Host: www.site.com
User-Agent: Opera
Accept: text/html, */*
Accept-Language: en-us
Accept: ISO-8859-1, utf-8
Connection: keep-alive

Response

Version Status Message status

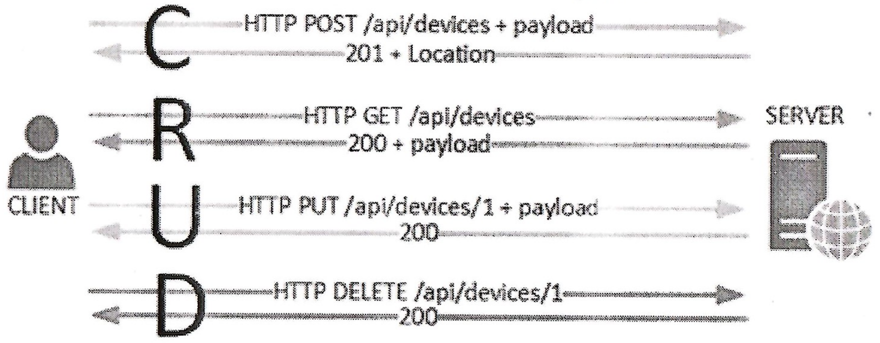
Header

```
HTTP/1.1 200 OK
Date: Sun, 21 Jul 2023
23:17 GTM
Server: IIS /6.0
Content-Type: text/html;
charset=UTF-8
Content-Length: 7686
```

blank line

Body

```
<html>
....
</html>
```



C
HTTP POST /api/devices + payload
201 + Location
R
HTTP GET /api/devices
200 + payload
U
HTTP PUT /api/devices/1 + payload
200
D
HTTP DELETE /api/devices/1
200
CLIENT
SERVER

- GET — getting a resource
- POST — resource creation
- PUT — resource update
- DELETE — deleting a resource

Methods		Request		Response	
		URL	Request body	Status	Response body
1	GET	Yes	No	Yes	Yes
2	PUT	Yes	Yes	Yes	No
3	POST	Yes	Yes	Yes	Yes
4	DELETE	Yes	No	Yes	No

WEB

+0.1 2025.9.18

String

```
class ACat {
string name;
public ACat(n){
  this.name=n;
 }
}
ACat mycat= new ACat("Barsik");
```

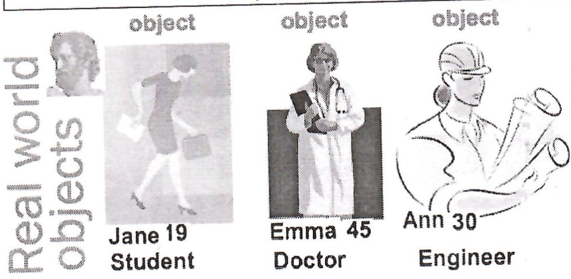
```
class ACat {
  constructor(n) {  -
    this.name = n;   //property
  }
}
mycat = new ACat("Barsik");
```

```
Say() {
  return "meou";
}
s =
 "My cat says <b>"
  +
 myCat.Say()+
 "</b> !"
```

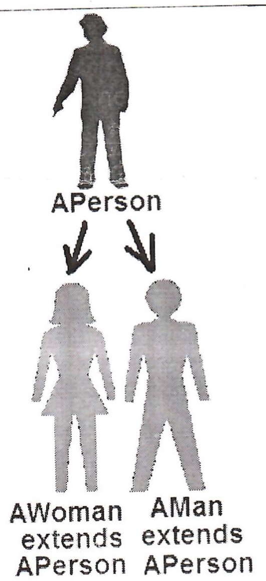
Error

```
<script>
class APet {
    Say() {
        alert("No");
    }
}
class ACat extends APet {
    Say() {
        return "Miou";
    }
}
var myPet =
  new ACat("Barsik");

 S = "My pet says  <b>"
     + myPet.Say()+
     "</b> !"
</script>
```

class Women To describe a Woman: name, age, job can do: eat, drink, sleep, walk, ...

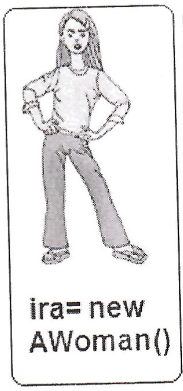


function **WashDishes()**

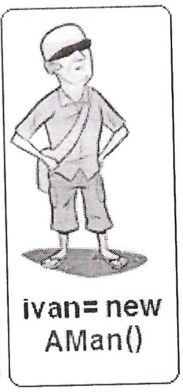


Men's version
wipe dry

Female version
wet with water



ira= new AWoman()



ivan= new AMan()

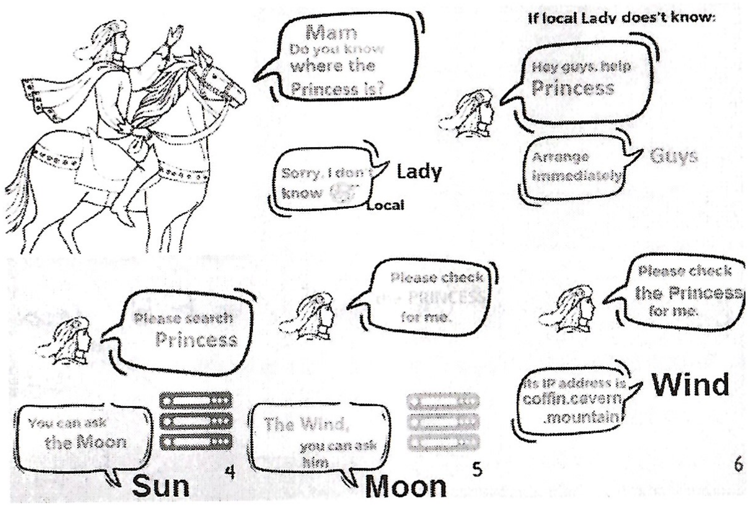
```
class AMan {
  WashDishes() {
    return 'wipe dry';
  }
}
```

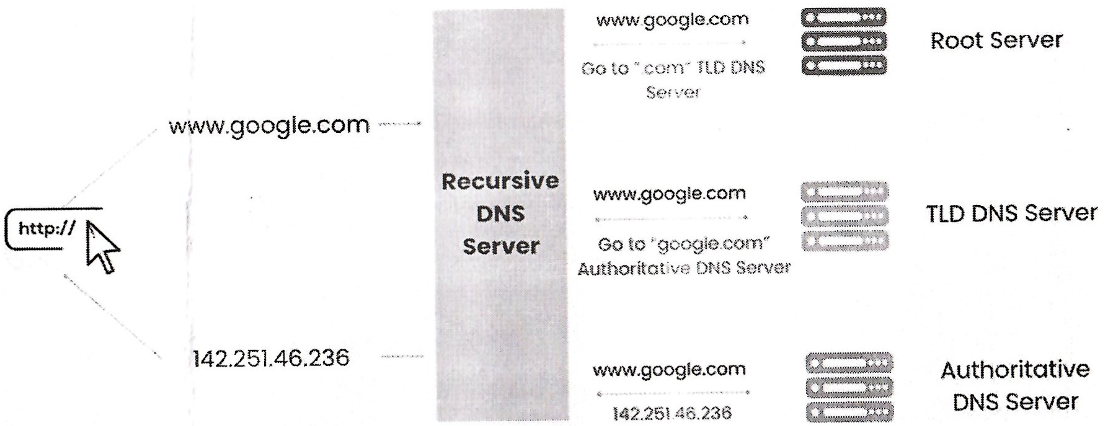
```
class AWomen {
  WashDishes() {
    return 'wet';
  }
}
```

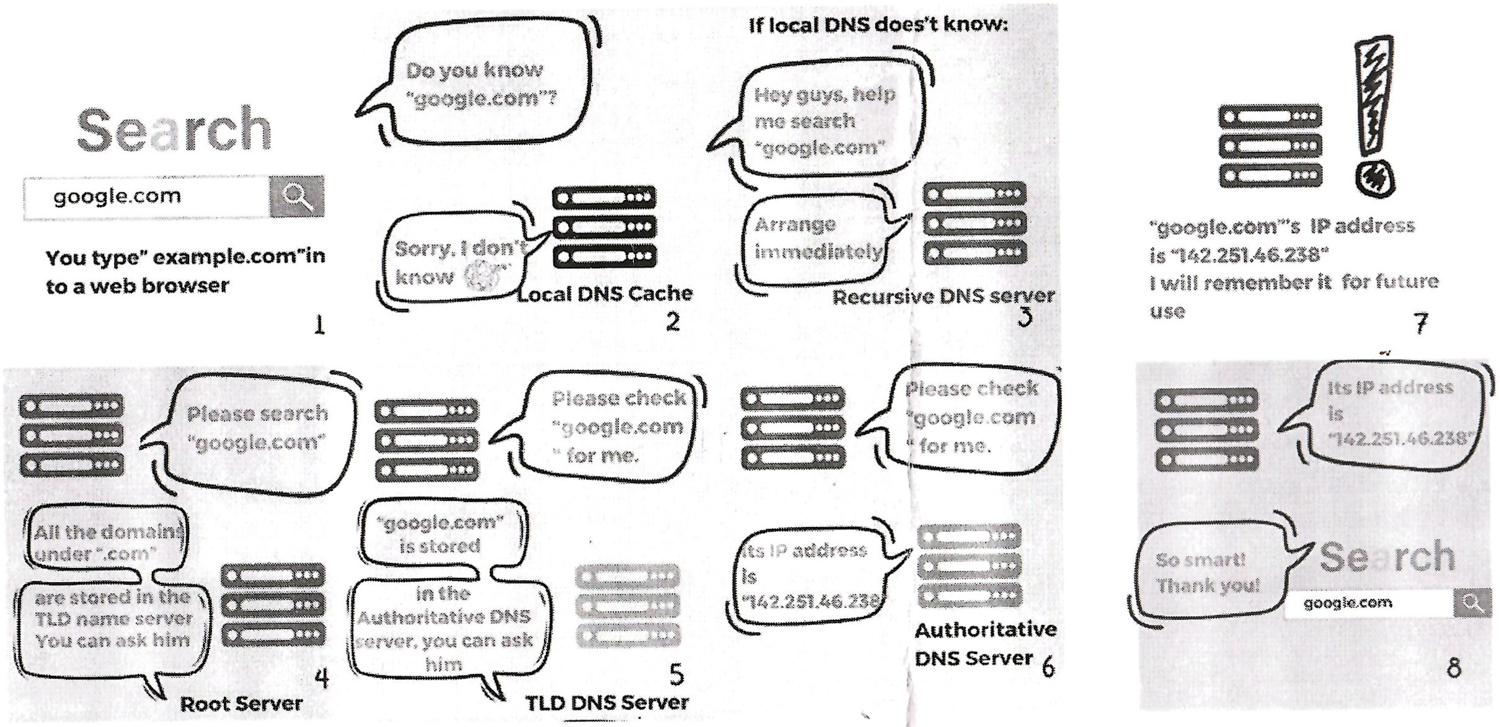
```
var family = [new AWomen(), new AMan()];
for(i=0;i<2;i++){ alert(" "+ family[i].WashDishes());}  //dry wet
```

```
women = new AWomen();  man = new AMan();  var family = [women,man];
```

WEB


If local Lady doesn't know:
Mam Do you know where the Princess is?
Sorry, I don't know
Lady
Local
Hey guys, help Princess
Arrange immediately
Guys
Please search Princess
You can ask the Moon
Sun
4
Please check PRINCESS for me.
The Wind, you can ask him
Moon
5
Please check the Princess for me.
Its IP address is coffin.cavern.mountain
Wind
6


http://
www.google.com
142.251.46.236
Recursive DNS Server
www.google.com
Go to ".com" TLD DNS Server
Root Server
www.google.com
Go to "google.com" Authoritative DNS Server
TLD DNS Server
www.google.com
142.251.46.236
Authoritative DNS Server

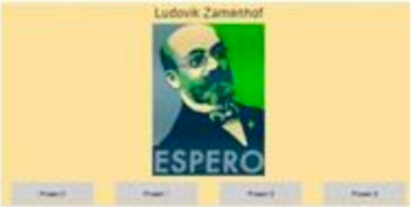

Search
google.com
You type" example.com"in to a web browser
1
Do you know "google.com"?
Sorry, I don't know
Local DNS Cache
2
If local DNS does't know:
Hey guys, help me search "google.com"
Arrange immediately
Recursive DNS server
3
Please search "google.com"
All the domains under ".com"
are stored in the TLD name server You can ask him
4
Root Server
Please check "google.com " for me.
"google.com" is stored
in the Authoritative DNS server, you can ask him
5
TLD DNS Server
Please check "google.com " for me.
Its IP address is "142.251.46.238"
Authoritative DNS Server
6
"google.com"'s IP address is "142.251.46.238"
I will remember it for future use
7
Its IP address is "142.251.46.238"
So smart! Thank you!
Search
google.com
8

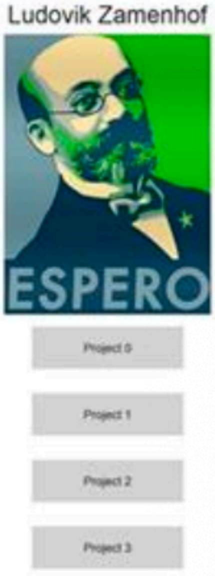
```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

<meta name="robots" content="noindex">

```
<style>
       #container {
         display: flex;
         flex-wrap: wrap;
         align: center;
       }
       #container > div {
         background-color: gray;
         font-size: 20px;
         margin: 20px;
         padding: 20px;
         width: 200px;
         text-align: center;
       }
</style>
```

```
  <body>
.................................
     <div id="container">
       <div>Project 0</a></div>
       <div>Project 1</div>
       <div>Project 2</div>
       <div>Project 3</div>
     </div>

  </body>
<html>
```



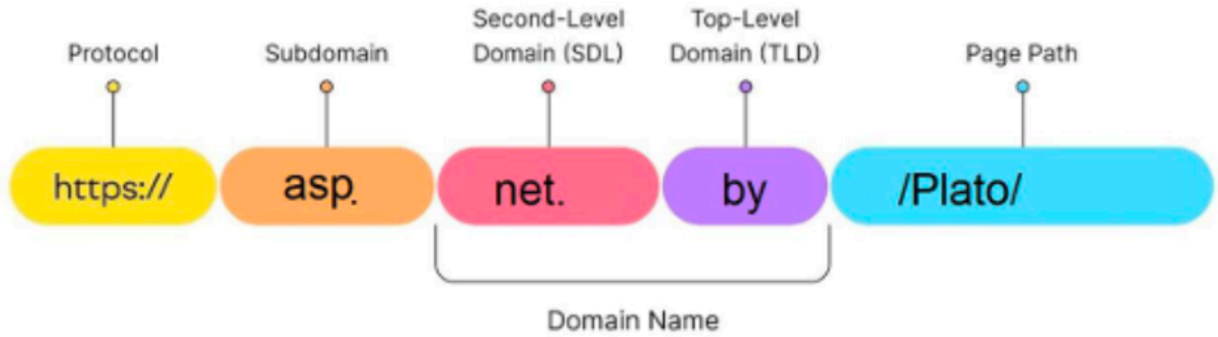


https://dot.net.by/esperanto/

```
<!-- Google Font -->

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap"
rel="stylesheet">

  <style>

    body {

      font-family: 'Poppins', sans-serif;

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
rel="stylesheet">

  <div class="buttons">
    <a href="project1.html" class="btn btn-primary btn-project">Project 1</a>

    @media (min-width: 768px) {
      .profile-image {
        width: 200px;
        height: 200px;
      }
      .name {
        font-size: 2.5rem;
      }
}
```

to enlarge photos on large screens from 150 to 200px

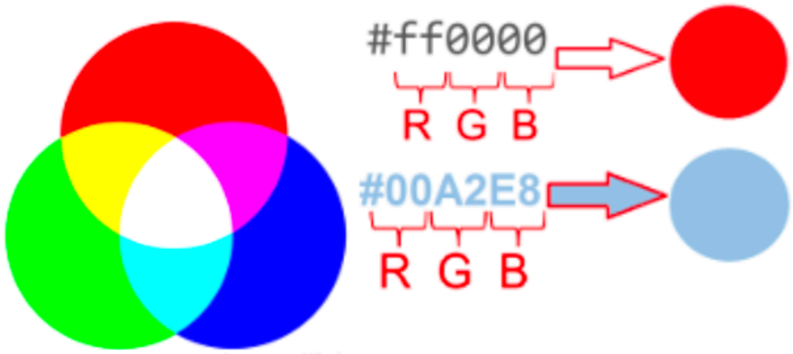
The same goes for fonts



R- Red

G- Green

B – Blue



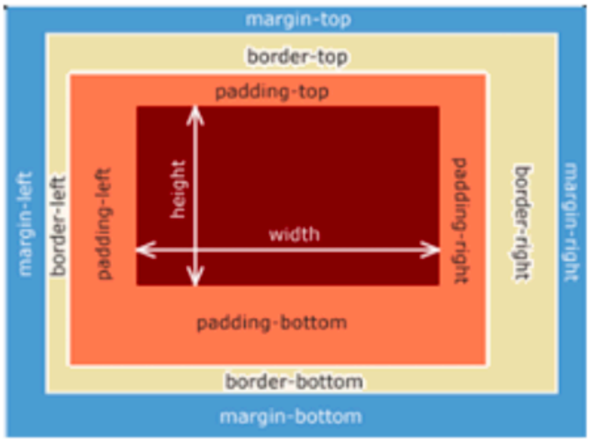
Intensity	0	1	1	0	1	0	0	1

8-digit cell

rgb(255,0,0), rgb(0,0,255), rgb(0,162,232), rgb(31,78,121)

rgb(100%,0%,0%), rgb(0%,0%,100%)

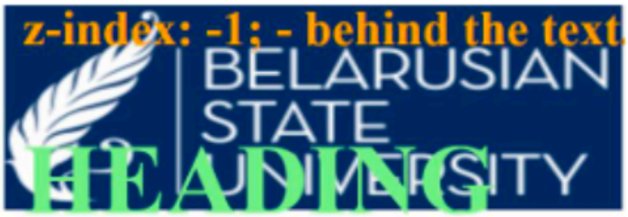
Box model

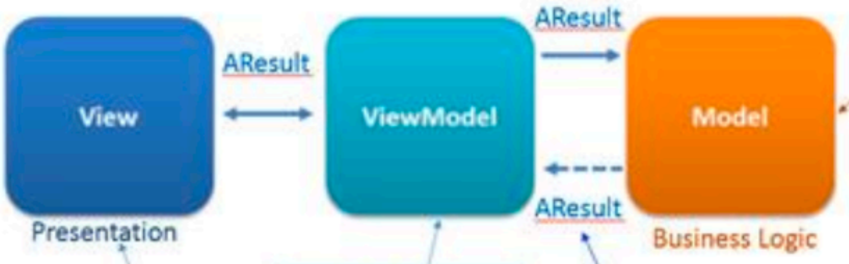


```
<style>
img {
  position: absolute;
  left: 2px;
  top: 10px;
  z-index: -1; // behind the text }
</style>
```

```
<style>
div
{
  color:white;
  background-color:brown;
  height:   250px;   width:    100px;
  padding:  20px  11px  10px  15px;
  border:   15px        solid   yellow;
  margin:  120px  10px  15px  20px;
}
</style>
```

<div><b>Box Model</b>



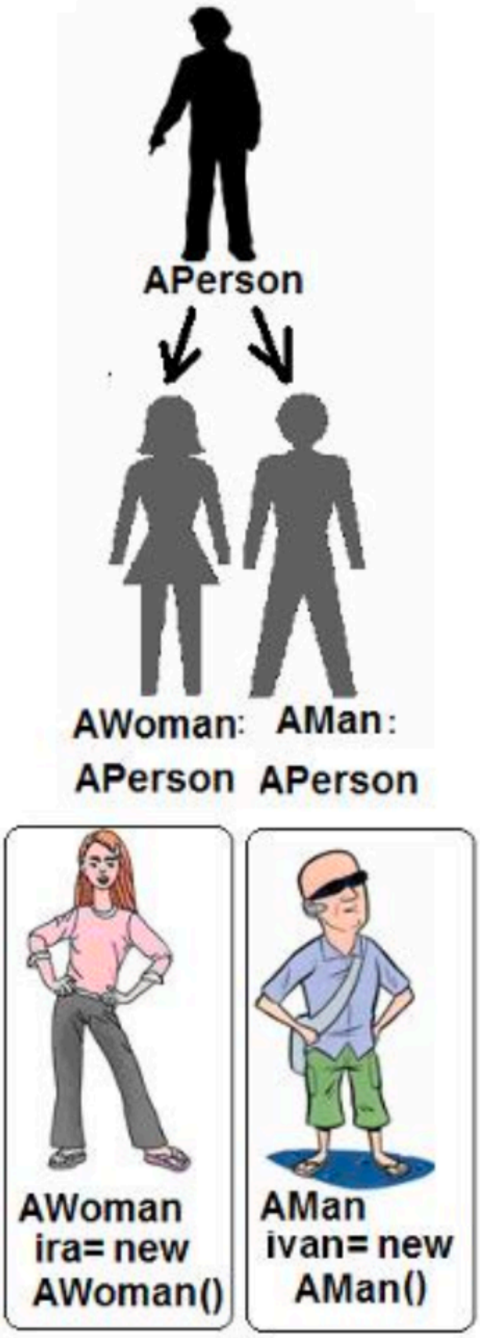


```
static AResult Model() {
        AResult result = new AResult();
        var single = new APerson();
       result.single += single.GetType().Name + ": " +
                                          single.WashDishes() + "";
        APerson woman = new AWomen();
        APerson man = new AMan();
        APerson[] family = new APerson[2];
        family[0] = woman;
        family[1] = man;
        for (int i = 0; i < 2; i++)
        {
           result.family += family[i].GetType().Name + ": " +
                        family[i].WashDishes() + "\n\r ";
        }
        return result;
    }
```

```
AResult result =
        Model();
View(result);

Console.ReadKey();
```

```
class AResult {
    internal string single ="";
    internal string family ="";
}
```

```
static void View( AResult result) {
 Console.WriteLine("Version for singles: \n\r "+result.single+"\n\r ");
 Console.WriteLine("Version for families: \n\r " + result.family + "\n\r ");
}
```



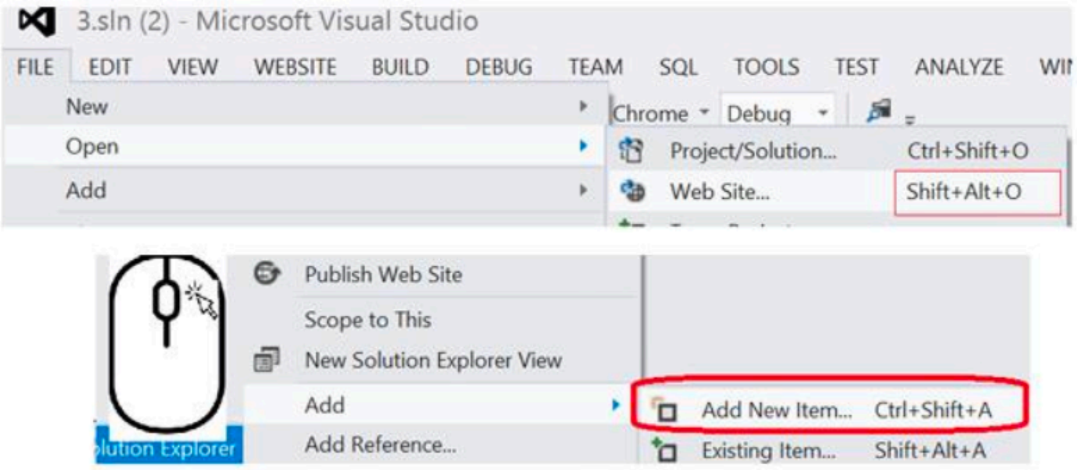
Men's version
wipe dry

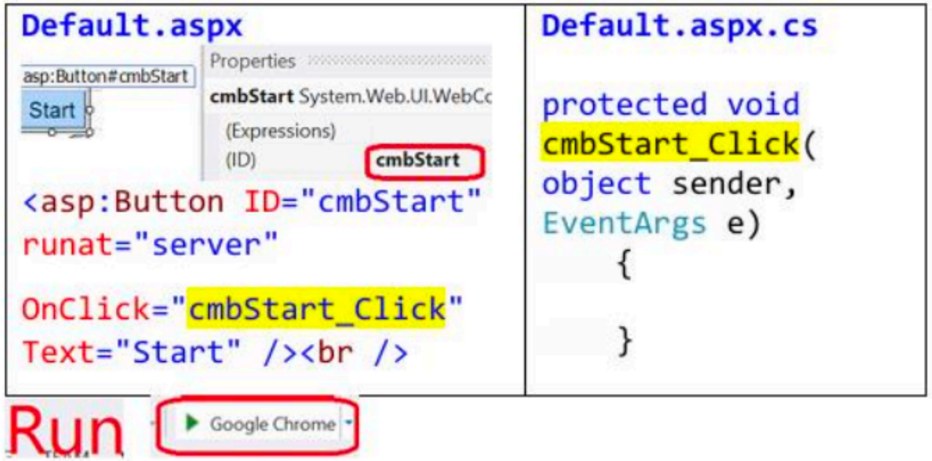
Female version
wet with water

```
class APerson {
    virtual public string WashDishes() {
        return "dry and wet";
    }
```

class AMan: APerson {	class AWoman: APerson {

override public
string WashDishes()
{
return "dry";
}
}
	override public
	string WashDishes()
	{
	return "wet";
	}
	}


3.sln (2) - Microsoft Visual Studio
FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE WI
New
Open
Add
Chrome
Debug
Project/Solution... Ctrl+Shift+O
Web Site... Shift+Alt+O
Publish Web Site
Scope to This
New Solution Explorer View
Add
Add Reference...
Add New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
lution Explorer



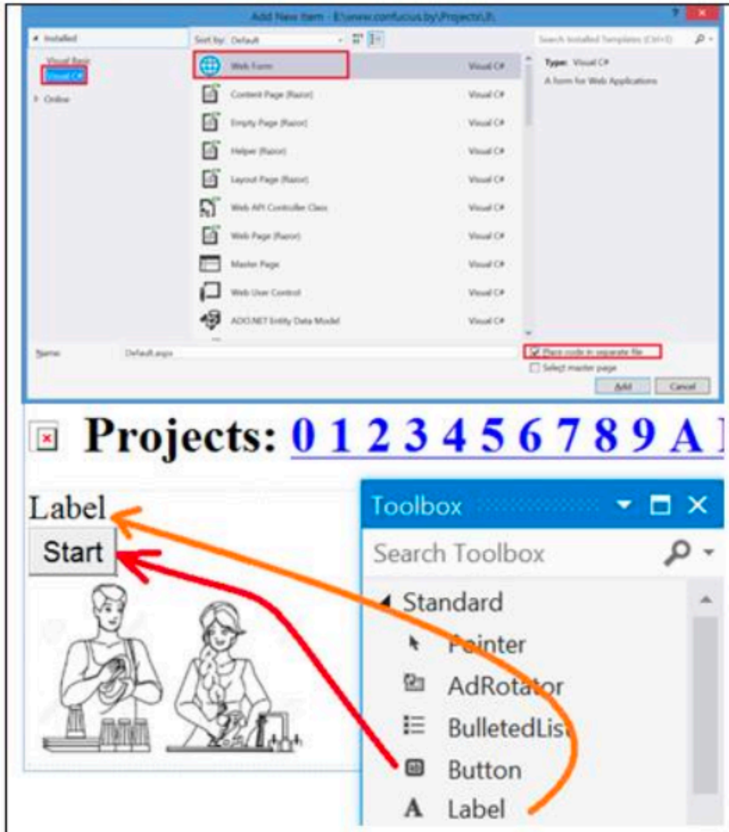
Default.aspx

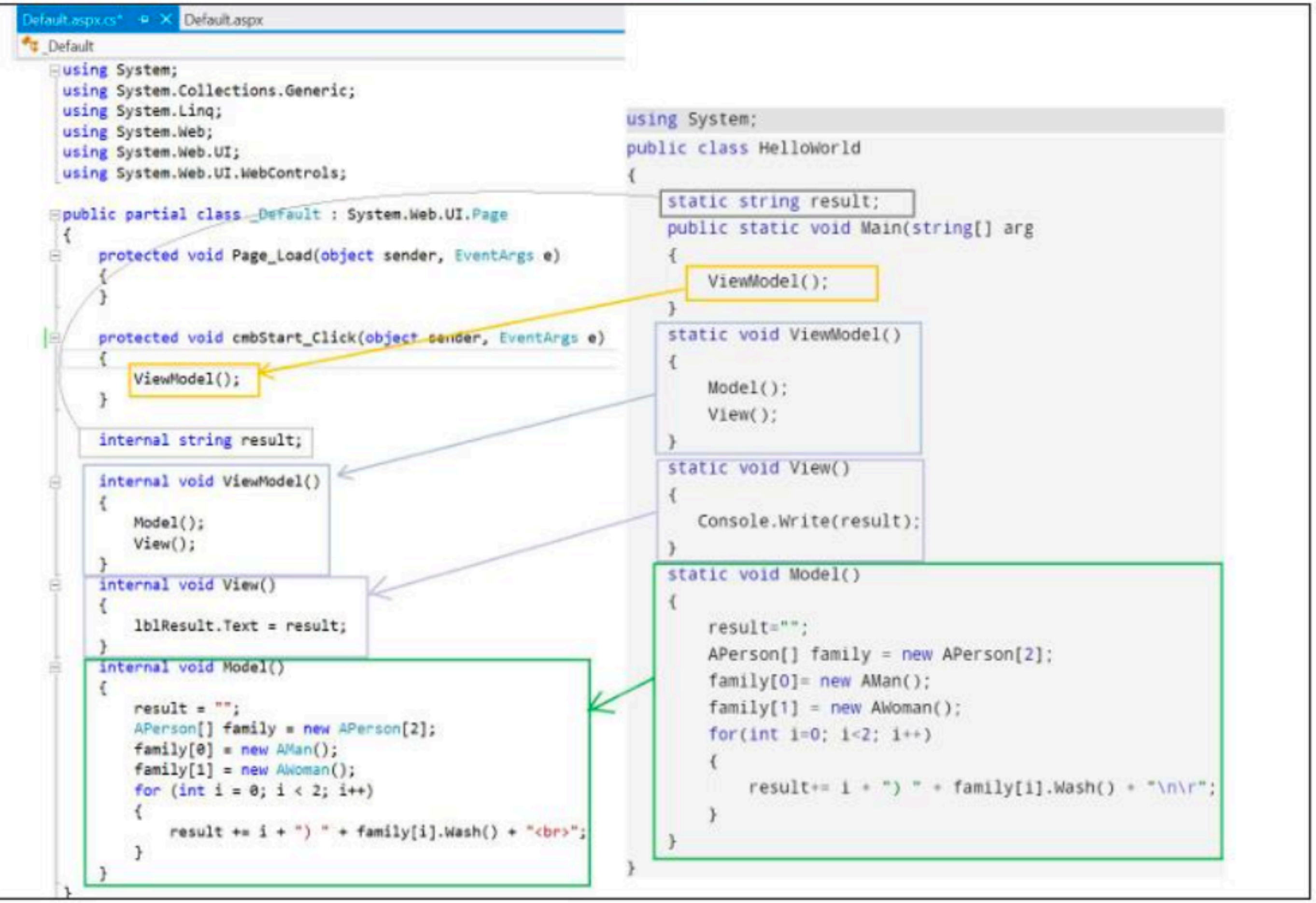
```
<asp:Button ID="cmbStart"
runat="server"
OnClick="cmbStart_Click"
Text="Start" /><br />
```

Default.aspx.cs

```
protected void
cmbStart_Click(
object sender,
EventArgs e)
    {

    }
```

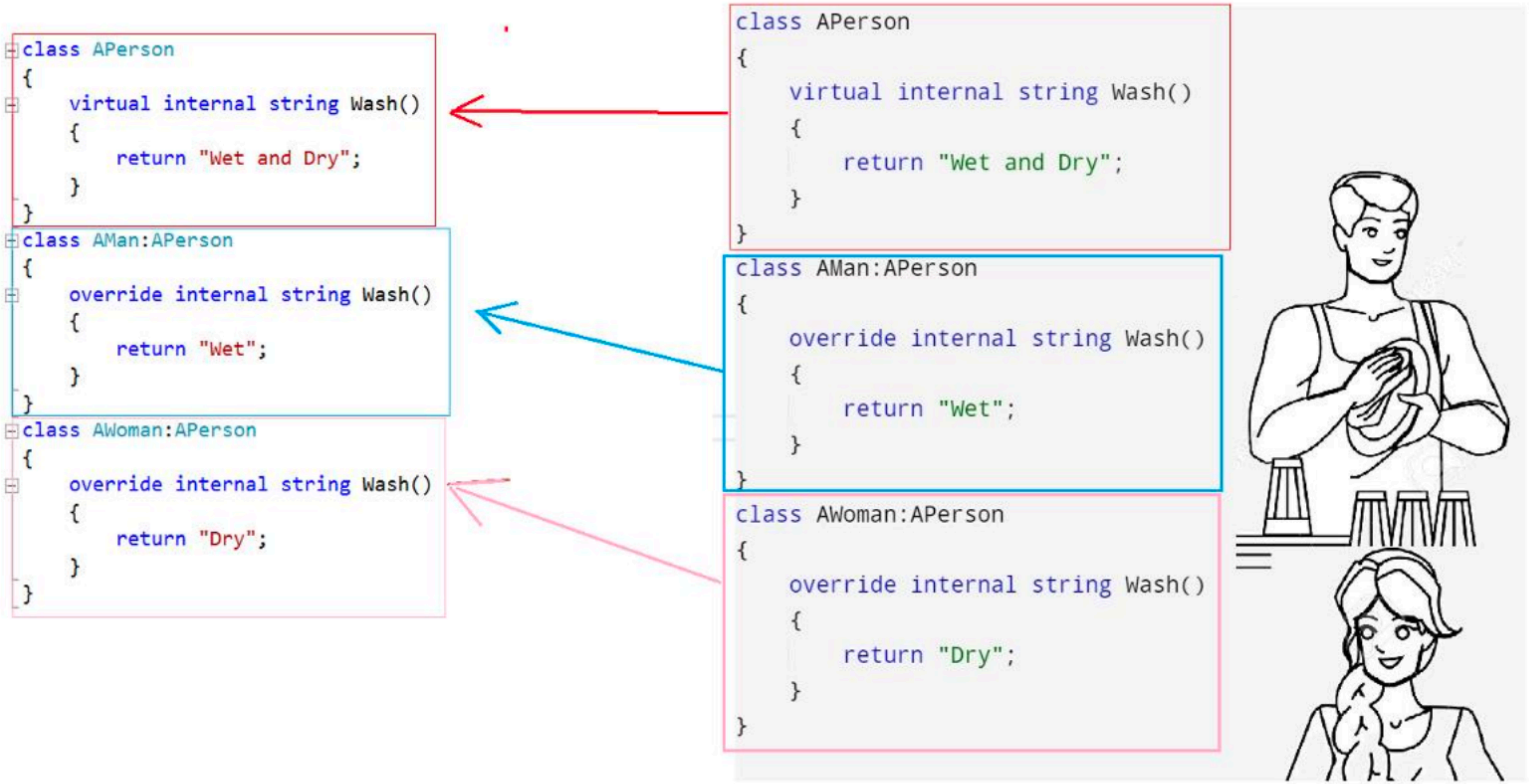
Run





```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void cmbStart_Click(object sender, EventArgs e)
    {
        ViewModel();
    }

    internal string result;

    internal void ViewModel()
    {
        Model();
        View();
    }
    internal void View()
    {
        lblResult.Text = result;
    }
    internal void Model()
    {
        result = "";
        APerson[] family = new APerson[2];
        family[0] = new AMan();
        family[1] = new AWoman();
        for (int i = 0; i < 2; i++)
        {
            result += i + ") " + family[i].Wash() + "<br>";
        }
    }
}
```

```
using System;
public class HelloWorld
{
    static string result;
    public static void Main(string[] arg
    {
        ViewModel();
    }
    static void ViewModel()
    {
        Model();
        View();
    }
    static void View()
    {
        Console.Write(result);
    }
    static void Model()
    {
        result="";
        APerson[] family = new APerson[2];
        family[0]= new AMan();
        family[1] = new AWoman();
        for(int i=0; i<2; i++)
        {
            result+= i + ") " + family[i].Wash() + "\n\r";
        }
    }
}
```



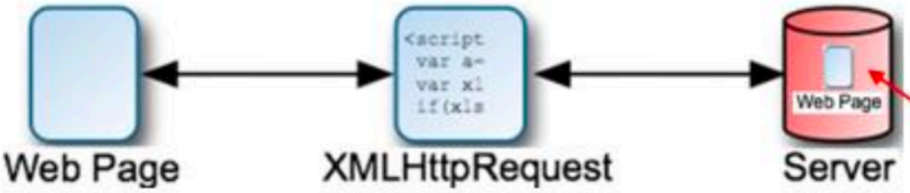
```
class APerson
{
    virtual internal string Wash()
    {
        return "Wet and Dry";
    }
}
class AMan:APerson
{
    override internal string Wash()
    {
        return "Wet";
    }
}
class AWoman:APerson
{
    override internal string Wash()
    {
        return "Dry";
    }
}
```

```
class APerson
{
    virtual internal string Wash()
    {
        return "Wet and Dry";
    }
}
class AMan:APerson
{
    override internal string Wash()
    {
        return "Wet";
    }
}
class AWoman:APerson
{
    override internal string Wash()
    {
        return "Dry";
    }
}
```

{"managers":[**<=JSON** vs **XML=>** < managers >
{ "firstName":"**Bill**",
"lastName":"**Gates**" },
{ "firstName":"**Mike**",
"lastName":"**Dell**" },
]}

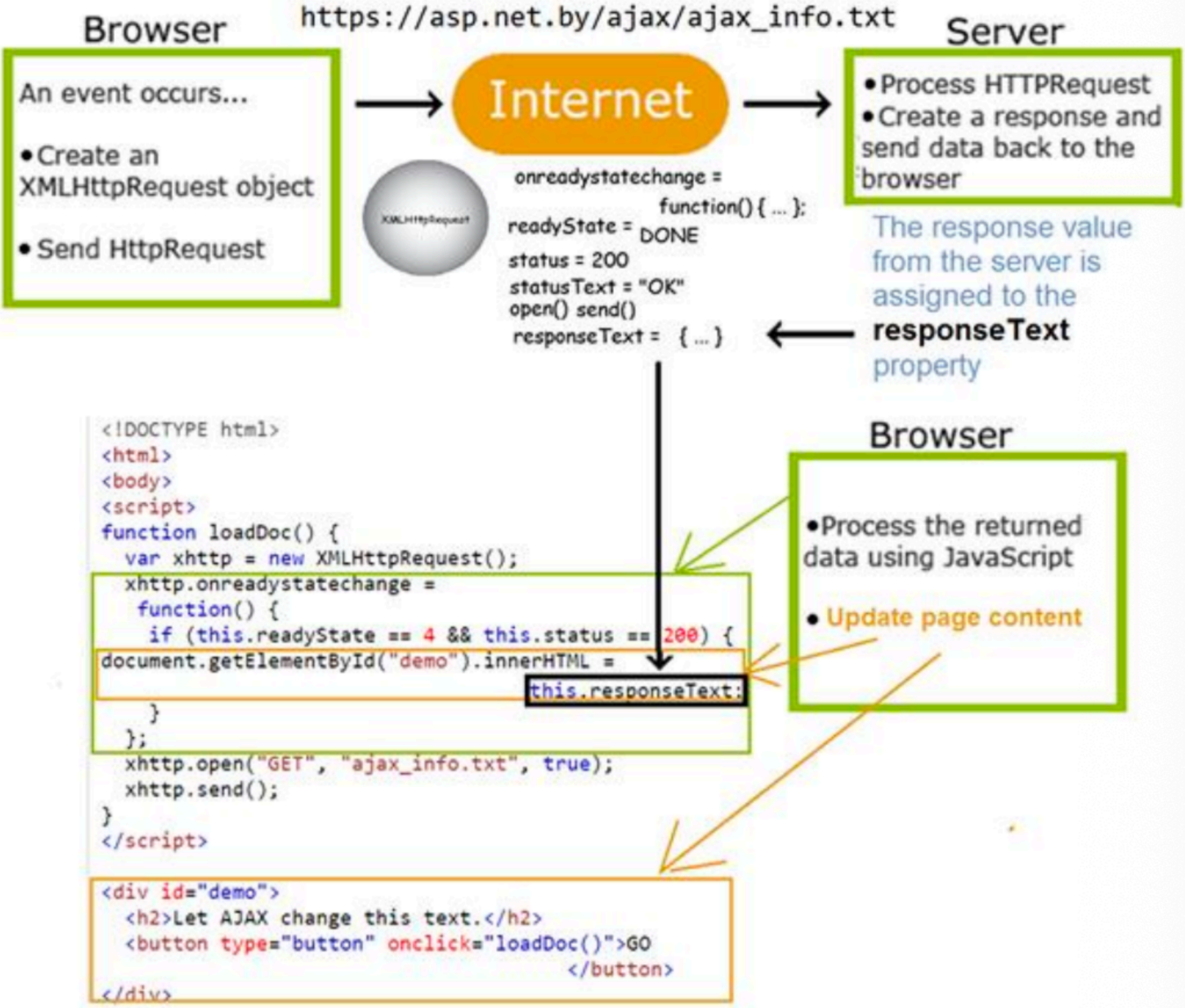
<manager> <firstName>**Bill**</firstName> <lastName>**Gates**</lastName> </manager>	
<manager> <firstName>**Mike**</firstName> <lastName>**Dell**</lastName> </manager>	

</managers>



For security reasons, modern browsers do not allow access across domains.

All must be located on the same server.



An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

- Process HTTPRequest
- Create a response and send data back to the browser

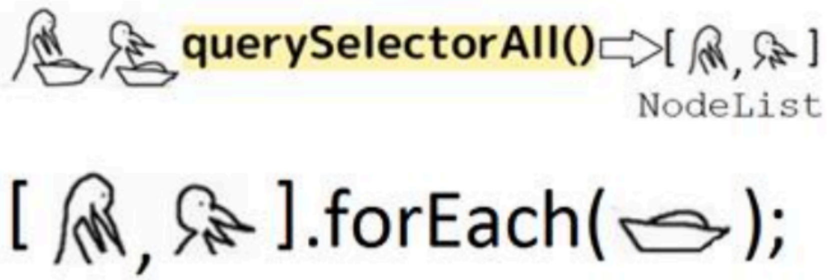
onreadystatechange = function() { ... };
readyState = DONE
status = 200
statusText = "OK"
open() send()
responseText = { ... }

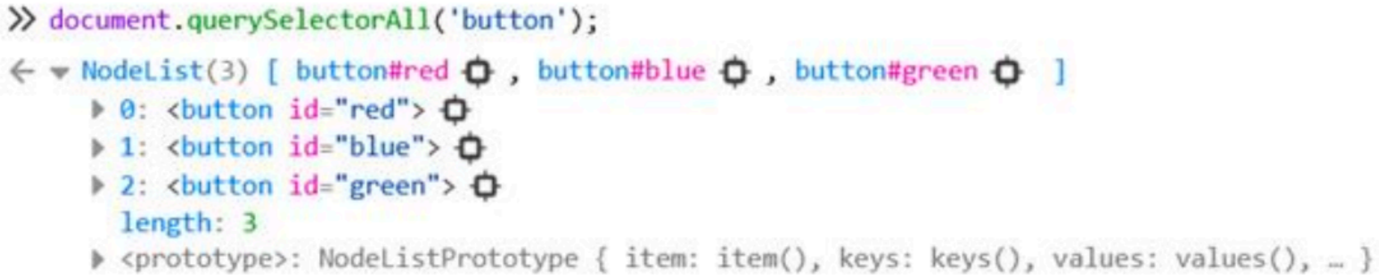
The response value from the server is assigned to the **responseText** property

- Process the returned data using JavaScript
- Update page content

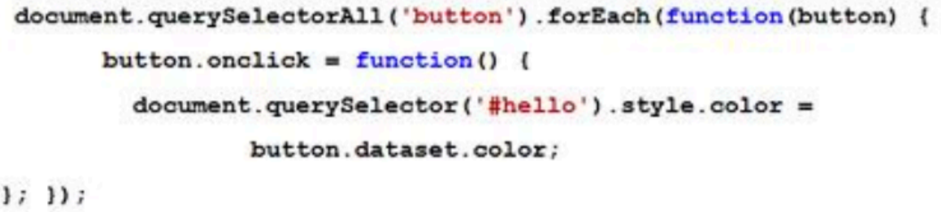
```
<!DOCTYPE html>
<html>
<body>
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange =
   function() {
    if (this.readyState == 4 && this.status == 200) {
document.getElementById("demo").innerHTML =
                                    this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

<div id="demo">
  <h2>Let AJAX change this text.</h2>
  <button type="button" onclick="loadDoc()">GO
                                   </button>
</div>
```

Сорока-Белобока кашу варила,
Деток кормила:
- Этому дала, - Этому дала,

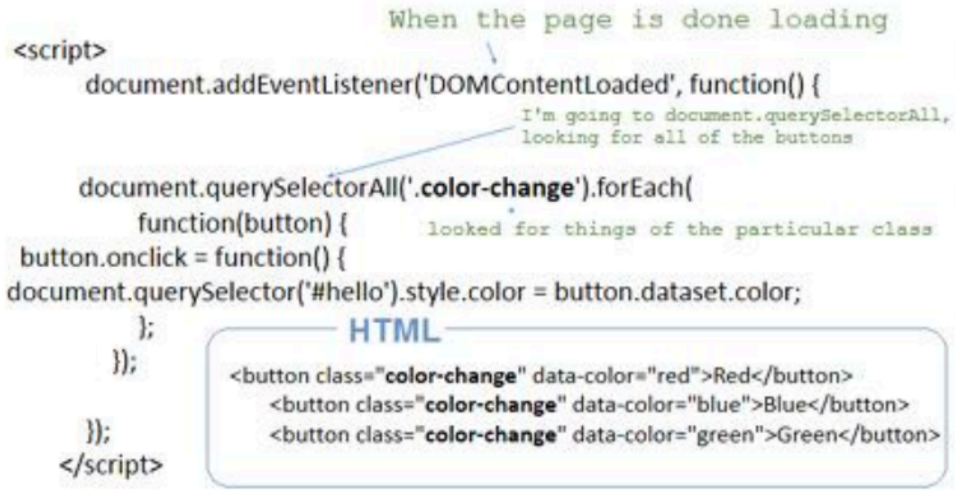




```
>> document.querySelectorAll('button');
← ▾ NodeList(3) [ button#red ⚙ , button#blue ⚙ , button#green ⚙  ]
     ▶ 0: <button id="red"> ⚙
     ▶ 1: <button id="blue"> ⚙
     ▶ 2: <button id="green"> ⚙
       length: 3
     ▶ <prototype>: NodeListPrototype { item: item(), keys: keys(), values: values(), … }
```



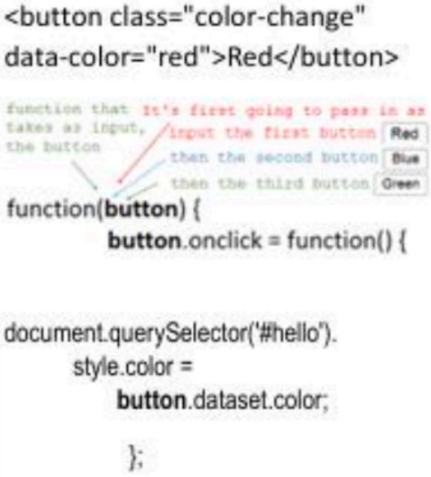
```
document.querySelectorAll('button').forEach(function(button) {
      button.onclick = function() {
        document.querySelector('#hello').style.color =
                button.dataset.color;
}; });
```



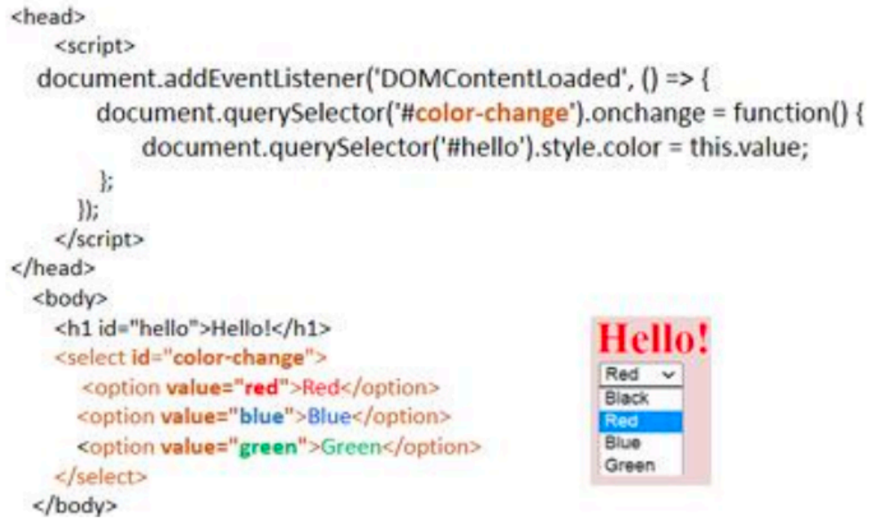
```
                                   When the page is done loading
<script>
     document.addEventListener('DOMContentLoaded', function() {
                                          I'm going to document.querySelectorAll,
                                          looking for all of the buttons

    document.querySelectorAll('.color-change').forEach(
          function(button) {          looked for things of the particular class
 button.onclick = function() {
document.querySelector('#hello').style.color = button.dataset.color;
          };
        });
        });
    </script>
```

HTML

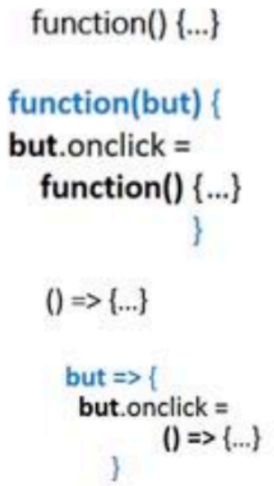
```
<button class="color-change" data-color="red">Red</button>
    <button class="color-change" data-color="blue">Blue</button>
    <button class="color-change" data-color="green">Green</button>
```



```
<button class="color-change"
data-color="red">Red</button>

function that   It's first going to pass in as
takes as input,  input the first button  Red
the button        then the second button  Blue
                  then the third button   Green
function(button) {
          button.onclick = function() {

document.querySelector('#hello').
     style.color =
          button.dataset.color;

        };
```



```
<head>
    <script>
  document.addEventListener('DOMContentLoaded', () => {
        document.querySelector('#color-change').onchange = function() {
             document.querySelector('#hello').style.color = this.value;
        };
      });
    </script>
</head>
  <body>
    <h1 id="hello">Hello!</h1>
    <select id="color-change">
       <option value="red">Red</option>
      <option value="blue">Blue</option>
      <option value="green">Green</option>
    </select>
  </body>
```



```
function() {...}

function(but) {
but.onclick =
   function() {...}
             }

() => {...}

 but => {
  but.onclick =
         () => {...}
     }
```